



# JAVA FOR BASE

GV: Bùi Quang Đăng





# Contents

1

Lịch học và quy định

2

Giới thiệu khóa học Java

3

Cài đặt môi trường lập trình

4

Làm việc với GIT cơ bản



# Giới thiệu Java Base



# Java™





# Giới thiệu Java for Base

## LỊCH HỌC VÀ QUY ĐỊNH



# Giới thiệu Java for Base

## ❖ Lịch học lớp Java for Base

- Học **2 buổi/ tuần** vào **T4, T6 từ 19h40**
- Địa điểm học: Tầng 2 số 20 ngõ 678 Đường Láng, Đống Đa, Hà Nội
- Giảng viên: **Bùi Quang Đăng**  
Mobile: **0987.232.936**  
Skype: buiquangdang  
Facebook: facebook/dangbq84  
Email: dangbq@Stanford.com.vn



# Giới thiệu Java for Base

## ❖ Quy định khóa học Java

- Các học viên được cung cấp tài liệu và slide bài giảng trước khi bắt đầu một chủ đề mới.
- Được cung cấp các tài khoản để sử dụng công cụ quản lý mã nguồn Git.
- Mỗi khi kết thúc buổi học, các học viên phải commit bài của mình lên **Git** trước khi ra về.
- Khi có việc bận không tham gia buổi học được, các học viên phải gọi điện cho giảng viên phụ trách trực tiếp để xin nghỉ.



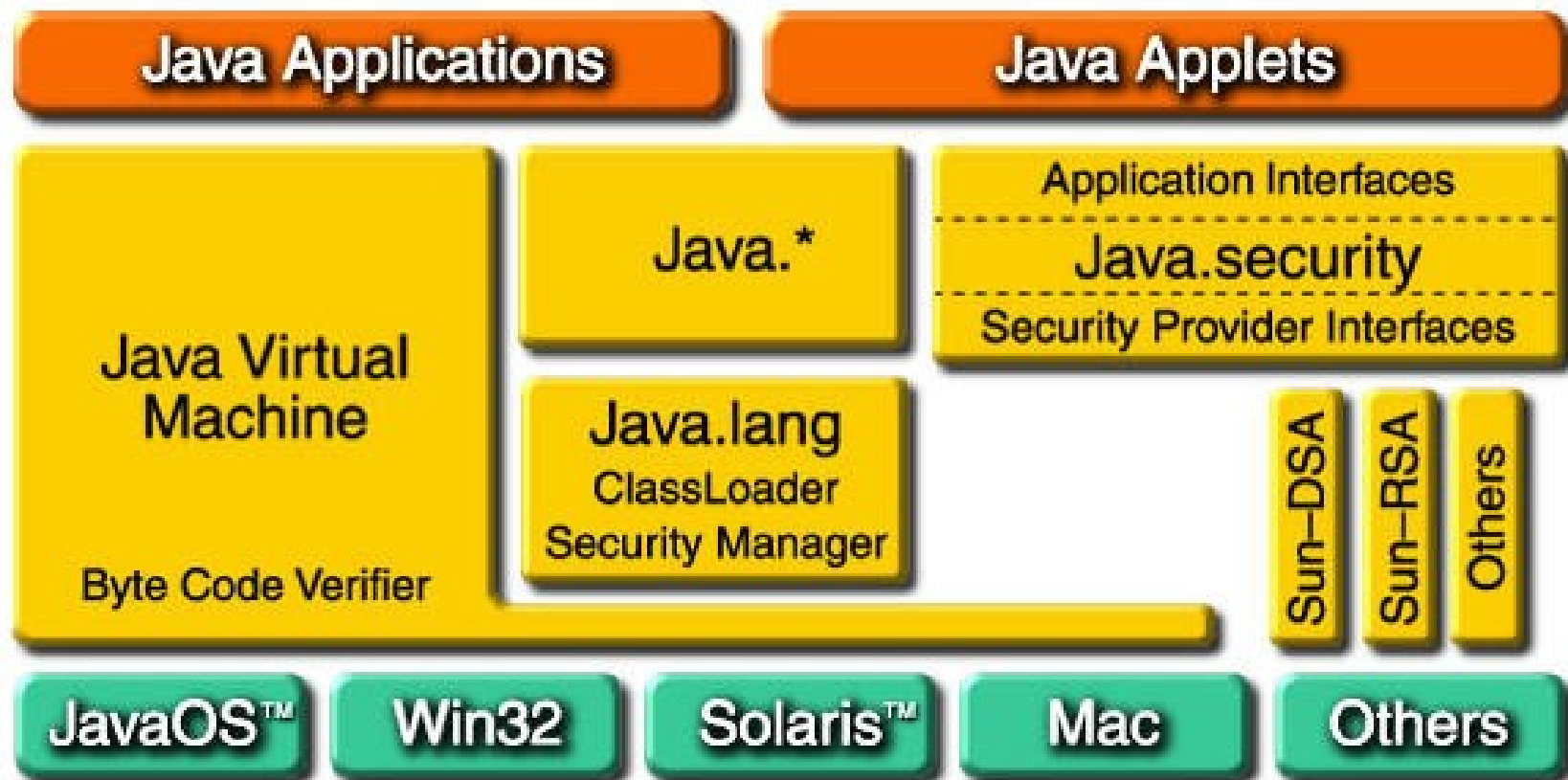


# Giới thiệu Java for Base

## GIỚI THIỆU KHÓA HỌC JAVA



# Giới thiệu Java for Base







# Giới thiệu Java for Base

## ❖ Nội dung khóa học Java

- Khóa học Java for Base tại Stanford cung cấp và trang bị cho các bạn 4 khối kiến thức quan trọng bao gồm:
  - Kiến thức nền tảng trọng tâm về Java (Java Core)
  - Lập trình giao diện ứng dụng trên Desktop
  - Làm việc với cơ sở dữ liệu MySQL Server
  - Project





# Giới thiệu Java for Base

## ❖ Nội dung chính khóa học Java

- Giới thiệu về Java, kiểu dữ liệu, toán tử và cấu trúc lập trình trong Java
- Làm việc với mảng, arraylist, list và collection generic trong Java.
- Làm việc Phạm vi truy cập, Hàm, thuộc tính
- Làm việc với lớp và đối tượng trong Java
- Làm việc với File trong Java
- Kế thừa & đa hình, Abstract và Interface trong Java





# Giới thiệu Java for Base

## ❖ Nội dung chính khóa học Java

- Làm việc với các thuật toán quan trọng về sắp xếp, tìm kiếm, max, min,...
- Làm việc với Java Desktop bằng Swing & AWT
- Làm việc với MySQL Server cơ bản
- Làm việc với MySQL bằng JDBC
- Projects





# Giới thiệu Java for Base

❖ Phương pháp học tập hiệu quả

STUDY





# Giới thiệu Java for Base

## ❖ Phương pháp học tập hiệu quả

- Mỗi chủ đề mới giảng viên sẽ dạy lý thuyết, giải thích và code demo để học viên hiểu ngay kiến thức được học. Sau đó tùy theo chủ đề sẽ có 1-3 buổi thực hành kế tiếp để học viên nắm chắc kiến thức mới được học
- Sau 4-5 tuần học, học viên sẽ làm bài kiểm tra đánh giá quá trình học của học viên.



# Giới thiệu Java for Base

## ❖ Phương pháp học tập hiệu quả

- Thực hiện đưa bài làm, thực hành của mình lên hệ thống tài nguyên được Stanford cấp để chuyên gia Stanford đánh giá, góp ý nâng cao kết quả trong quá trình học
- Làm bài kiểm tra trắc nghiệm và bài thực hành tổng hợp
- Thực hiện làm project cuối khóa học





# Giới thiệu Java for Base

## ❖ Kết quả đạt được sau khóa học Java

- Nắm chắc kiến thức nền tảng về ngôn ngữ lập trình Java phục vụ cho công việc, nghiên cứu chuyên sâu
- Làm việc và sử dụng thành thạo công cụ lập trình Eclipse, Netbean, công cụ quản lý mã nguồn Git và các đối tượng controls hay sử dụng trên Desktop
- Sử dụng và thiết kế cơ sở dữ liệu cũng như làm việc với MySQL Server bằng JDBC
- Xây dựng Project cuối khóa hoàn thiện một sản phẩm sử dụng các kỹ thuật đã học
- Trình bày và tổng kết khóa học





# Giới thiệu Java for Base

## ❖ Quyền lợi dành cho bạn

- Được cấp chứng chỉ khi có điểm tổng kết từ 5 điểm trở lên.
- Tham gia hội nhóm chia sẻ kiến thức về lập trình
- Được đặt cách qua 2 vòng đầu tuyển dụng tham gia dự án **CodePlus** – Tuyển thực tập lập trình đào tạo thành lập trình viên chuyên nghiệp của Stanford:
  - *Đào tạo miễn phí, Tham gia dự án của công ty*
  - *Có ngay 1 năm kinh nghiệm làm việc*
  - *Giới thiệu việc làm sau chương trình*
  - *Trở thành nhân viên chính thức, Tham gia hoạt động đào tạo*





# Giới thiệu Java for Base

## CÀI ĐẶT MÔI TRƯỜNG LẬP TRÌNH



# Giới thiệu Java for Base

## ❖ Cài đặt môi trường làm việc

- NetBeans, Eclipse,...
- Git
- Database Management (MySQL, SQL Server,...)
- Notepad++



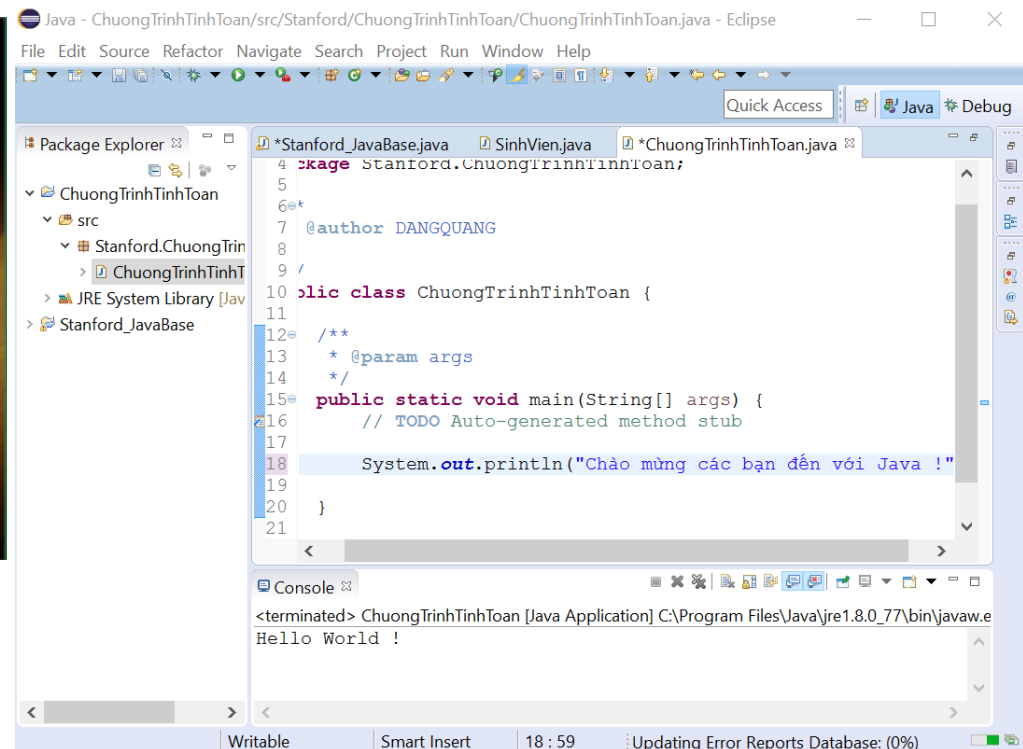


# Giới thiệu Java for Base

## ❖ Cài đặt môi trường làm việc



- **Eclipse:** Công cụ sử dụng để lập trình ngôn ngữ Java





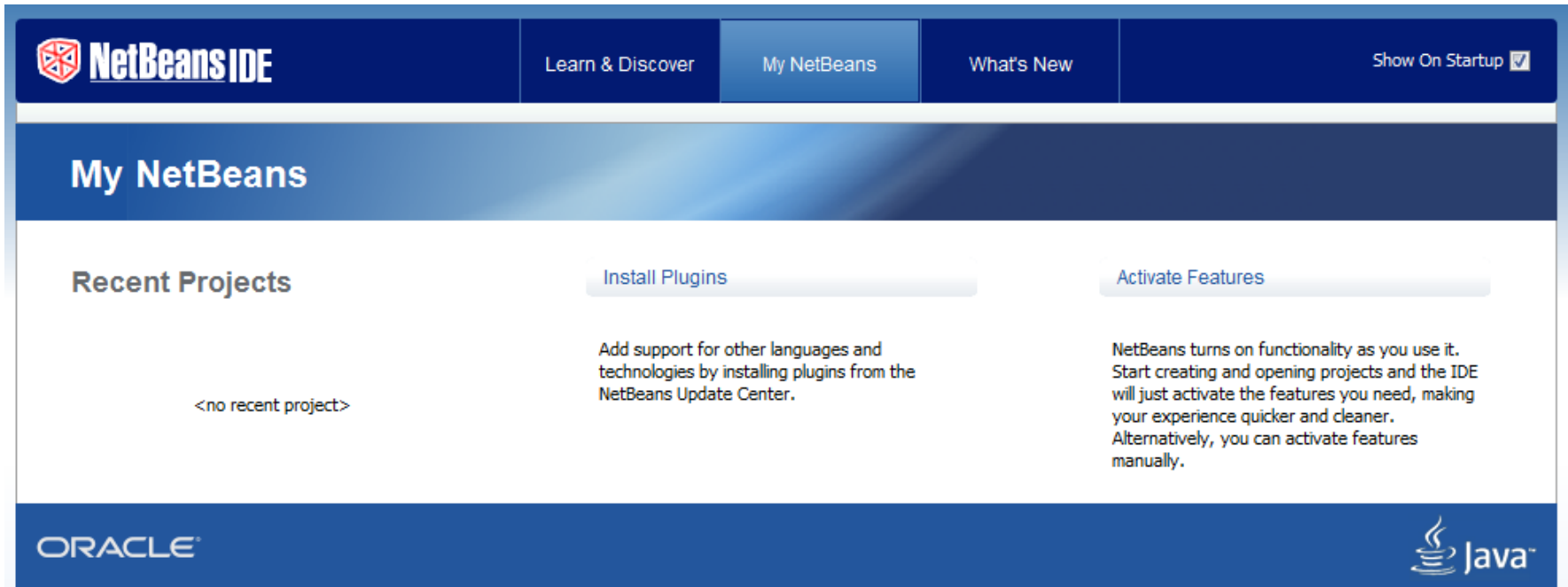
# Giới thiệu Java for Base



**NetBeans**

## ❖ Cài đặt môi trường làm việc

- **NetBeans:** Công cụ phát triển ứng dụng Java Desktop





# Cài đặt môi trường làm việc

## ❖ Các công cụ lập trình, tiện ích

- MySQL Server



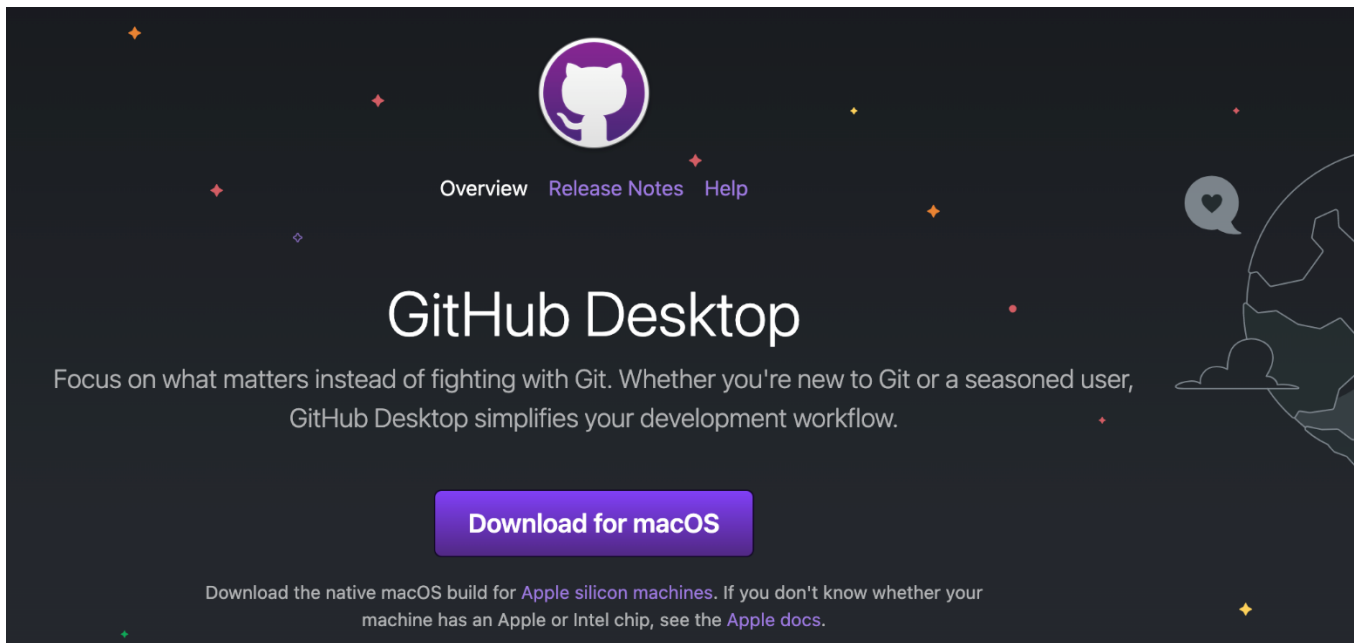
- MySQL server: <https://dev.mysql.com/downloads/mysql/>
- MySQL Workbench: <https://dev.mysql.com/downloads/workbench/>



# Cài đặt môi trường làm việc

## ❖ Cài đặt môi trường làm việc

- **Git:** Công cụ quản lý source code

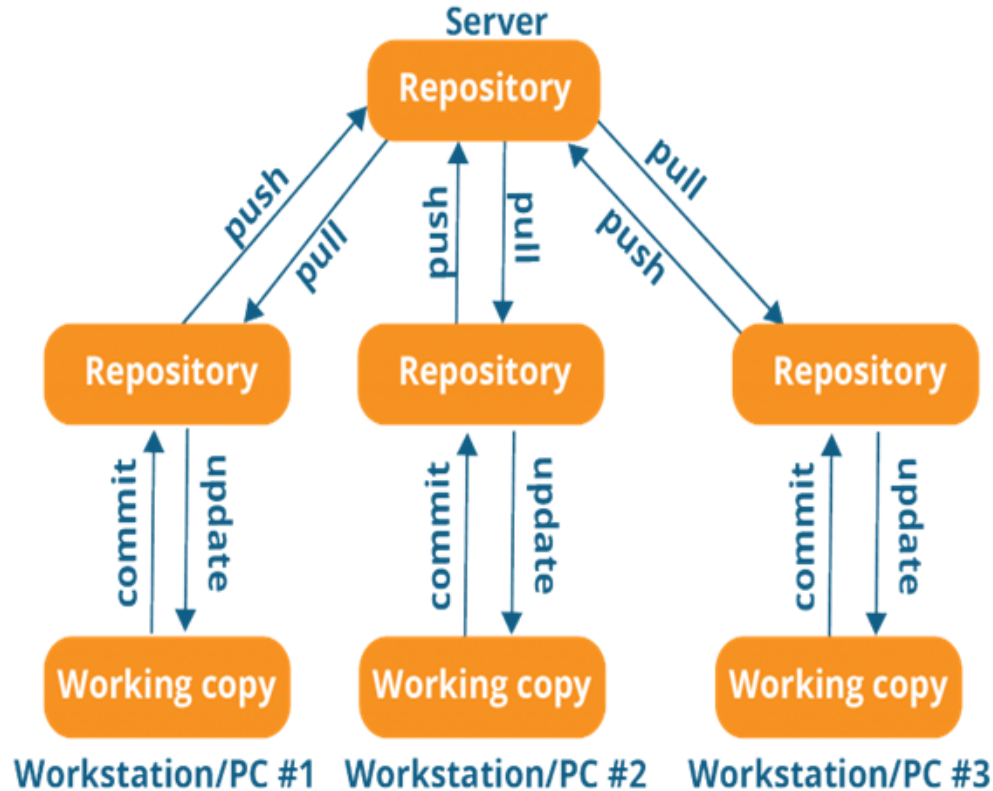


- Địa chỉ tải bộ cài: <https://desktop.github.com/>

# Cài đặt môi trường làm việc

## ❖ Công cụ quản lý mã nguồn Git

Distributed version control system





# Cài đặt môi trường làm việc

## ❖ Hướng dẫn cài đặt GIT

### ■ macOS:

- Thực hiện mở terminal sau đó chạy lệnh sau qua brew:

```
brew install git
```

- Cài hỗ trợ giao diện đồ họa:

```
brew install git-gui
```

- Hoặc cài công cụ GitHub Desktop, tải bộ cài: [tai đây](#)





# Cài đặt môi trường làm việc

## ❖ Hướng dẫn cài đặt GIT

### ■ Windows:

- Bạn có thể sử dụng một số công cụ sau:
  - Git for Windows: [tại đây](#)
  - GitHub Desktop: [tại đây](#)
  - TortoiseGit: [tại đây](#)
- Thực hiện tải bộ cài theo đường dẫn trên về máy tính sau đó thực hiện cài đặt.



# Cài đặt môi trường làm việc

## ❖ Cài đặt môi trường làm việc

### ■ Máy server của Stanford:

- LAN: **192.168.1.200**
- Tài khoản mạng LAN: **hocvien/123456**

### ■ Địa chỉ tài nguyên của lớp:

- Git: <https://github.com/StanfordHV/J042401CB.git>
- Sử dụng tài khoản email đăng ký để kết nối vào tài nguyên của lớp
- Địa chỉ nhóm FB:

<https://www.facebook.com/groups/193471849128825>



# Cài đặt môi trường làm việc

## ❖ Cài đặt môi trường làm việc

- **Git:** Công cụ quản lý source code
  - **Git clone:** Sử dụng lần đầu tiên để kết nối với server chứa mã nguồn theo địa chỉ cung cấp.
  - **Git pull:** Cập nhật tài nguyên mới nhất từ server về máy tính của từng thành viên trong nhóm.
  - **Git push:** Người lập trình sử dụng để đưa tài nguyên lên server khi được yêu cầu.



# Java for Base

## GIỚI THIỆU VỀ GIT



# Working with GIT

## ❖ GIT là gì ?

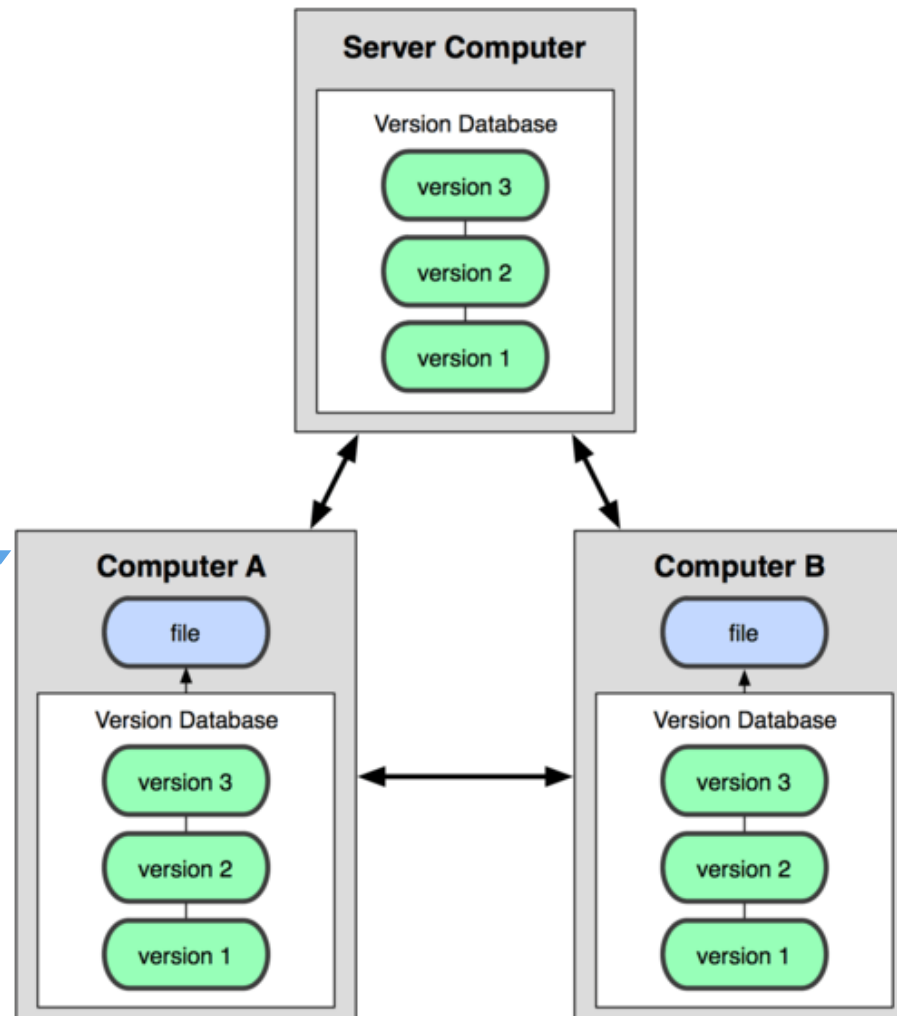
- Git là một hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS), nó là một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay.
- **Git** cung cấp cho mỗi lập trình viên kho lưu trữ (**repository**) riêng chứa toàn bộ lịch sử thay đổi.



# Working with GIT

## ❖ GIT là gì ?

Quản lý các  
phiên bản trên  
GIT

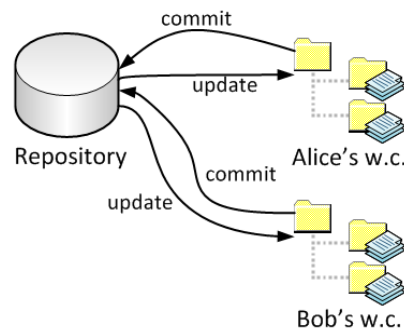




# Working with GIT

## ❖ VCS (Version Control System) là gì ?

- **VCS** là viết tắt của Version Control System là hệ thống kiểm soát các phiên bản phân tán mã nguồn mở. Các VCS sẽ lưu trữ tất cả các file trong toàn bộ dự án và ghi lại toàn bộ lịch sử thay đổi của file. Mỗi sự thay đổi được lưu lại sẽ được và thành một version (phiên bản).





# Working with GIT

## ❖ VCS (Version Control System) là gì ?

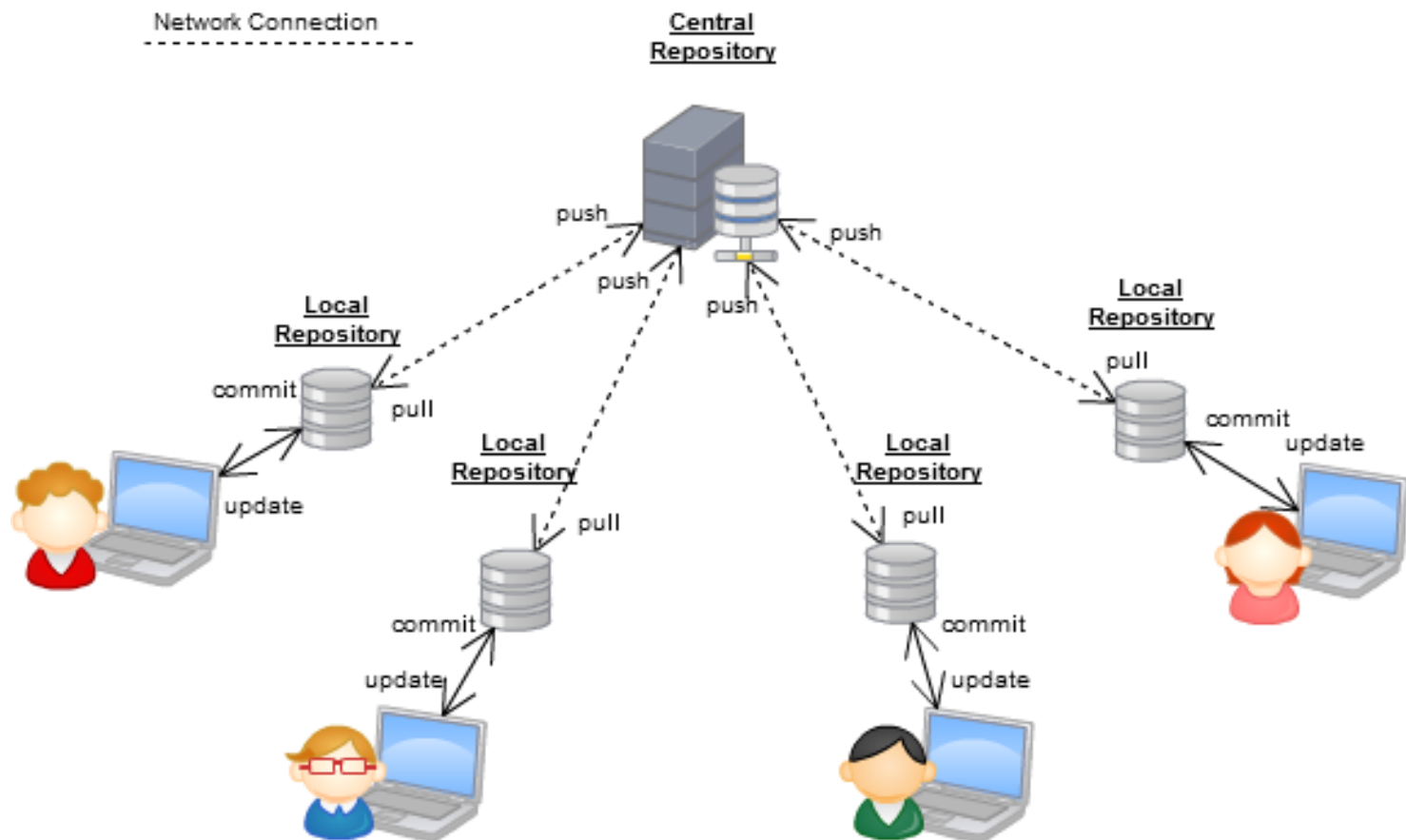
- **VCS** là hệ thống giúp lập trình viên có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (clone) từ một kho chứa mã nguồn (repository), mỗi thay đổi vào mã nguồn trên local sẽ có thể ủy thác (commit) rồi đưa lên server nơi đặt kho chứa chính.





# Working with GIT

## ❖ VCS (Version Control System) là gì ?





# Working with GIT

## ❖ Ý nghĩa của VCS là gì ?

- Lưu lại lịch sử các version của bất kỳ thay đổi nào của dự án. Giúp xem lại các sự thay đổi hoặc khôi phục (revert) lại sau này.
- Việc chia sẻ code trở nên dễ dàng hơn, lập trình viên có thể để public cho bất kỳ ai, hoặc private chỉ cho một số người có thẩm quyền có thể truy cập và lấy code về.
- Git là một VCS ghi nhớ lại toàn bộ lịch sử thay đổi của source code trong dự án. Lập trình sửa file, thêm dòng code tại đâu, xóa dòng code ở hàng nào...đều được Git ghi nhận và lưu trữ lại.



# Java for Develoer

## LÀM VIỆC VỚI GIT CƠ BẢN



# Working with GIT

## ❖ Các lệnh hay sử dụng trong GIT

### ■ **git init:**

- Khởi tạo 1 git repository 1 project mới hoặc đã có.
- **Cú pháp:**

```
git init
```

- Di chuyển vào thư mục cần làm việc sau chạy lệnh trên



# Working with GIT

## ❖ Các lệnh hay sử dụng trong GIT

### ■ git remote:

- Để check remote/source bạn có hoặc add thêm remote
- Cú pháp:

```
git remote add [URL Repository]
```

– Ví dụ:

- » `git remote add https://github.com/StanfordHV/CS092101CB.git`  
thêm thư mục hiện thời kết nối với remote.
- » `git remote` để kiểm tra và liệt kê



# Working with GIT

## ❖ Các lệnh hay sử dụng trong GIT

### ■ **git clone:**

- Copy một repository từ remote source về máy tính của bạn
- Cú pháp:

```
git clone <url repository>
```

– Ví dụ: `git clone https://github.com/StanfordHV/CS102101DEV.git`



# Working with GIT

## ❖ Các lệnh hay sử dụng trong GIT

- **git clone:** Sử dụng trên GitHub desktop

Clone a Repository

GitHub.com

GitHub Enterprise

URL

Repository URL or GitHub username and repository  
( hubot/cool-repo )

git@github.com:StanfordHV/PY102101CB.git

Local Path

/Volumes/DATA/Stanford/Courses/Python/PY102101/BaiGiang-Gi

Choose...

Cancel

Clone



# Working with GIT

## ❖ Các lệnh hay sử dụng trong GIT

### ■ **git status:**

- Để check trạng thái của những file bạn đã thay đổi trong thư mục làm việc. Ví dụ: Tất cả các thay đổi cuối cùng từ lần commit cuối cùng.
- **Cú pháp:**

```
git status
```

- Di chuyển thư mục cần kiểm tra





# Working with GIT

## ❖ Các lệnh hay sử dụng trong GIT

### ■ **git add:**

- Thêm thay đổi đến stage/index trong thư mục làm việc để có thể commit và đưa lên repository.
- **Cú pháp:**

**git add “Tên file hoặc thư mục cần đưa”**

– Ví dụ: git add “demo.txt”



# Working with GIT

## ❖ Các lệnh hay sử dụng trong GIT

### ■ **git commit:**

- commit nghĩa là một hành động để Git lưu lại một snapshot của các sự thay đổi trong thư mục làm việc. Và các tập tin, thư mục được thay đổi đã phải nằm trong Staging Area. Mỗi lần commit nó sẽ được lưu lại lịch sử chỉnh sửa của code kèm theo tên và địa chỉ email của người commit.
- Ngoài ra trong Git bạn cũng có thể khôi phục lại tập tin trong lịch sử commit của nó để chia cho một branch khác, vì vậy bạn sẽ dễ dàng khôi phục lại các thay đổi trước đó.



# Working with GIT

## ❖ Các lệnh hay sử dụng trong GIT

### ■ **git commit:**

- commit nghĩa là một hành động để Git lưu lại một snapshot của các sự thay đổi trong thư mục làm việc.
- **Cú pháp:**

```
git commit -m “Ghi chú cho thao tác”
```

– Ví dụ: `git commit -m “Thêm mới file demo.txt”`

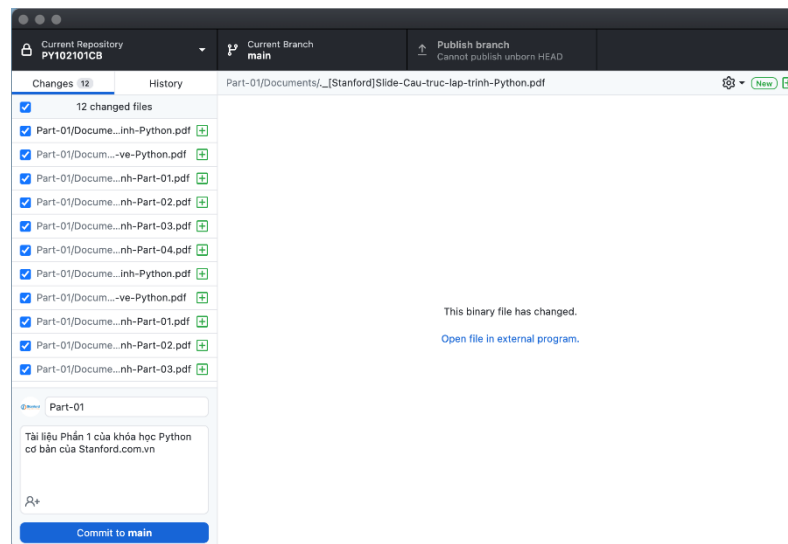


# Working with GIT

## ❖ Các lệnh hay sử dụng trong GIT

### ■ git commit: Trên GitHub Desktop

- Copy thư mục, file vào thư mục đã clone từ git về
- Thực hiện commit trên công cụ GitHub Desktop:

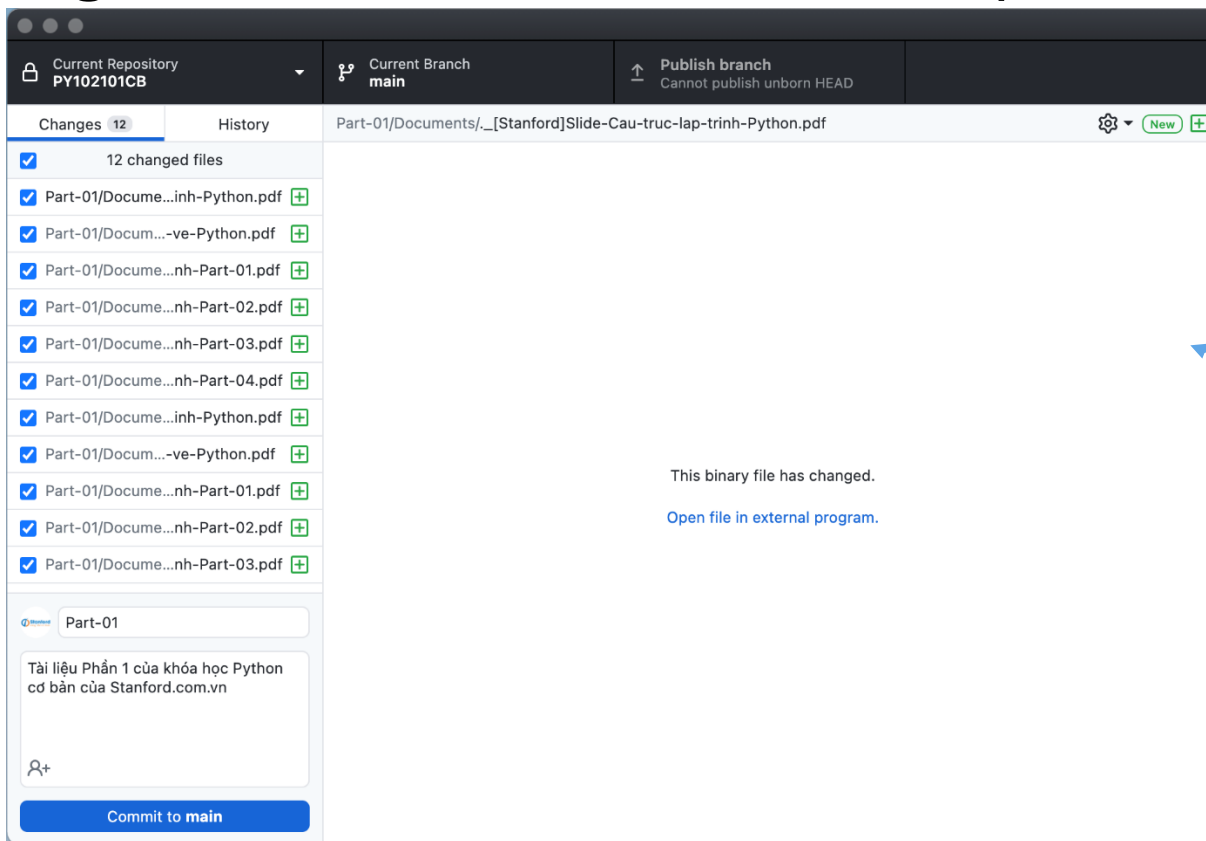




# Working with GIT

## ❖ Các lệnh hay sử dụng trong GIT

### ■ git commit: Trên GitHub Desktop



Nhấn Commit  
để thêm vào git



# Working with GIT

## ❖ Các lệnh hay sử dụng trong GIT

### ■ git push:

- Đưa code thay đổi của mình lên remote hay máy chủ chứa.
- Cú pháp:

```
git push origin <Tên nhánh cần đưa lên>
```

– Ví dụ: git push -u origin main

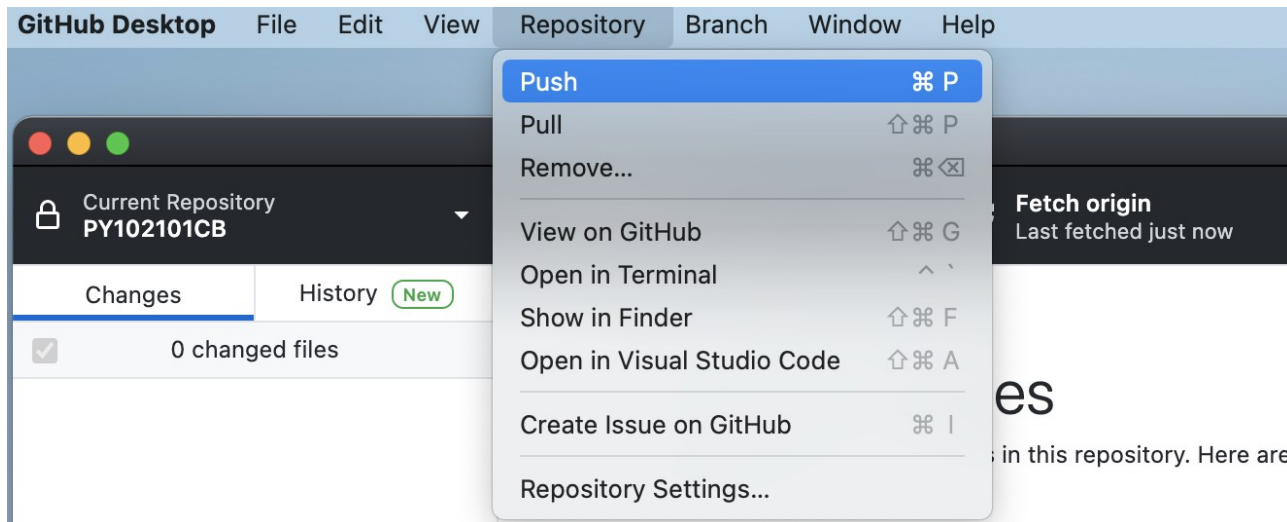


# Working with GIT

## ❖ Các lệnh hay sử dụng trong GIT

### ■ git push:

- Trên GitHub Desktop nhấn vào nút Push hoặc vào menu Repository => Push



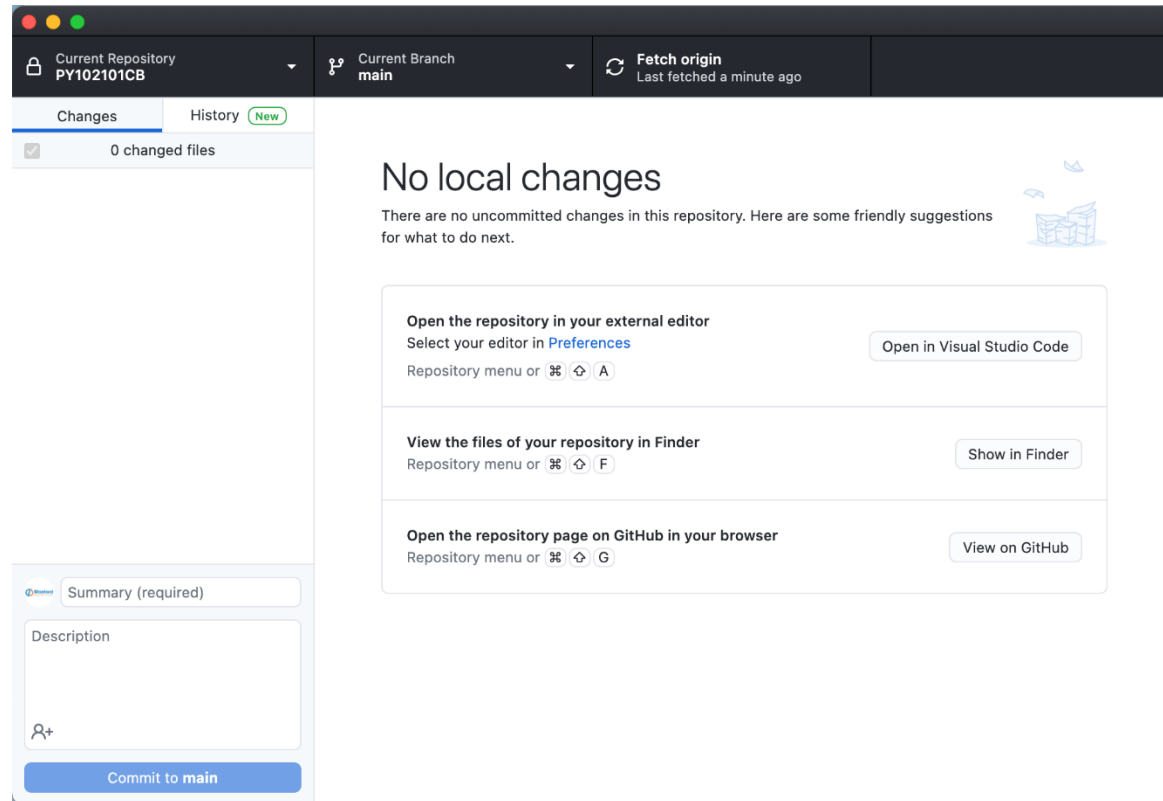


# Working with GIT

## ❖ Các lệnh hay sử dụng trong GIT

### ■ git push:

- Giao diện khi các file, dữ liệu đã được đưa lên server trên GitHub Desktop:







# Working with GIT

## ❖ Các lệnh hay sử dụng trong GIT

### ■ git pull:

- Cập nhật thông tin mới từ remote hay máy chủ về máy tính của mình.
- **Cú pháp:**

```
git pull
```

– Ví dụ: git pull



# Thank You !

[www.stanford.com.vn](http://www.stanford.com.vn)